Messaging in Difficult Environments

Delay Tolerant Networking as a Generalized Messaging Service

Kevin Fall
kfall@intel.com
Intel Research Berkeley
Oct/Nov, 2004
http://www.dtnrg.org

***Abstract*— Electronic mail, text messaging, instant messaging, and network news are all important applications used for both business and personal communications. At present, each form their own individual, separate networks. In meeting user demand for the capability to utilize multiple such networks anywhere, a variety of devices have become available (including laptops, advanced cell phones, and PDAs) equipped with a multitude of network interfaces. As such devices are called upon to operate in increasingly diverse environments, the performance of the underlying network infrastructure is reflected in the application performance, and generally the overall user experience.**

Originating as an architecture to support high-latency asynchronous interplanetary communications, the Delay Tolerant Networking (DTN) architecture provides secure, reliable messaging (somewhat similar to e-mail) across heterogeneous, failure-prone networks and a standard method to incorporate proxies where necessary to interconnect radically heterogeneous communications systems. Unlike present Internet electronic mail and instant messaging systems, DTN incorporates a framework for dynamic routing and fragmentation, including the capability to utilize both scheduled routing and *data mules* (mobile store-and-forward nodes that essentially move messages by employing physical transportation) and to recover efficiently from network interruption. In this brief overview paper, we discuss the problems encountered in implementing communications in difficult-to-reach and disruption-prone environments, the major components of the DTN architecture designed to combat these problems, and how DTN compares with other current messaging systems. We then contemplate how DTN could be used as an underlying general mechanism for supporting these systems. For systems incorporating human *presence* as an important aspect, and thereby operating most effectively when latency is low, the architecture provides fewer benefits, but still helps in the areas of interoperability and short-term disruption.

I. Introduction

Asynchronous messaging services, especially text messaging and electronic mail, have increased in popularity. Arguably, one of their attractive properties is their naturally *asynchronous* nature. In particular, a user does not need to be diverted from other tasks in order to participate in a conversation. Conversely, a growing number of users are also willing to be diverted under some circumstances, and they are using instant messaging (IM), which provides a more real-time oriented messaging service. Some users specifically prefer IM systems for certain messages, because such messages are to be acted on immediately or discarded (a semantic not offered by text messaging or e-mail).

A difficulty arises when trying to map these differing user requirements onto the heterogeneous collections of networks that are being called upon to support them. At a minimum, networks consisting of the Internet, cellular telephone system, and Bluetooth/USB/serial wired personal area networks may all participate as component networks in an overall internetwork. At present, these networks are only moderately compatible with each other and only occasionally interconnected.

A. Towards a New Messaging Architecture

In reconciling the wide range of user demands and underlying support networks, it seems apparent that any new architecture should excel in handling asynchronous messaging and should handle instant messages reasonably well, without excess delay. Furthermore, the communication service offered should consist of a least common denominator "basic" offering that can be made to operate on and among a very wide range of existing networks. When additional network capabilities are available end-to-end, they can be offered to applications as an option.

Asynchronous communication can be supported in either connected networks, or occasionally-connected (intermittent) networks. The converse is not true: synchronous communication can only be reasonably supported in networks that can reasonably be considered to be always connected. Synchronous communication has been a (loose) assumption of the Internet's design. Although it does not offer hard guarantees on delivery performance (cf. bounded delay or delay distribution), there is a common understanding that Internet routing will not delay packets "very long." If it does, application problems become common due to short timeouts or other issues.

If only intermittently-connected network infrastructure is required for a significant fraction of communications, the cost of building the infrastructure could be dramatically reduced [3]. In some extreme cases, a synchronous communication structure is inherently infeasible anyhow (e.g. for neworks that never maintain a contemporaneous end-to-end connection). In such circumstances, unconventional data distribution methods (such as data mules [13]) may become attractive.

B. Challenged Internetworks

Networks that cannot easily support the Internet's performance or architectural assumptions have been called *chal-*

lenged internetworks [8]. We now summarize the types of performance and environmental concerns that arise when dealing with internetworks of this kind:

- **network infrastructure**: Lack of infrastructure may force users to be intermittently connected. Devices disconnected for long periods of time may require significant persistent storage to hold the traffic demand until connectivity resumes.
- **interruption**: Scheduled down time, interference, or environmental hostility may cause the interruption of otherwise-operable communication links. When scarcity of power makes communication costly and therefore infrequent, achieving efficient utilization of communication opportunities becomes very important.
- **heterogeneity**: Challenged networks cannot generally be assumed to be running a common set of protocols in each node, thereby requiring some additional mechanisms to support interoperable communication. Any such approach will need to accommodate a high degree of variation in naming, addressing, rate control, and routing approaches. In particular, support for proxies that can be placed at convenient points of interconnection in the network topology is of significant importance.

These properties are *extrinsic* factors affecting the design of an architecture aimed at providing communication in stressed or difficult-to-reach locations. That is, they are characteristics of the operating environment that must be accommodated by any architecture. In addition, there are several *intrinsic* features that are desirable in their own right, but must be synthesized by the implementation of a design instance of the architecture itself. These include the following:

- **security** The scarcity of available bandwidth in challenged internetworks dictates that some form of authentication and access control is required, and that its enforcement should be applied as early as possible along the chain of routing elements used to deliver a message. Otherwise, precious link resources may be used to carry messages that will only be discarded when they reach their intended recipient(s).
  While many protocols have been proposed in the security literature that can provide authentication and access control at multiple points in the network, most of them do not tolerate long latencies. In particular, protocols that require multiple round-trip data exchanges or multiple client-server interactions to achieve their security will not be appropriate for challenged networks that suffer from frequent long-term disconnection.
- **reliability** Challenged internetworks are expected to not only be limited in bandwidth and connectivity, but may also have high error rates. When link error rates are sufficiently large to cause packet loss, two methods are commonly used to correct the problem: retransmission and various coding techniques (e.g. erasure coding). For networks where link error rates are high, end-to-end retransmission is unlikely to be effective and should instead be implemented using some hop-by-hop approach. A 'hop' refers to a hop among the agents responsible for message forwarding, at whatever layer it may be implemented. This is true for today's e-mail routing agents (MTAs), for example, where messages are retransmitted some small maximum number of times before failure is declared.
- **handling of user preferences** Allowing users to express some aspects of the importance of their messages can be of significant benefit, both to the users and to the network infrastructure which supports them. Two natural aspects of such user preferences are a sense of relative message priority (e.g. this message should be delivered ahead of that other one, if possible), and a notion of useful life (e.g. this message is only useful for the next five minutes; if it cannot be delivered in that amount of time, it might as well be discarded). In most of today's messaging systems, neither of these aspects are handled very effectively. As for priority, relative indicators are typically used only to signal the recipient (e.g. red exclamation marks), and timeliness can only be specified to be at one extreme or the other (e.g. in email and text messaging, the recipient decides when to discard; in IM, the message is discarded if the recipient is not immediately available).

II. Delay Tolerant Networking

The recently-proposed Delay Tolerant Networking architecture [6] offers one approach for solving the problems of challenged internetworks. The DTN architecture supports data mules, scheduled and opportunistic network connectivity, enhanced end-to-end reliability with a hop-by-hop store and forward mechanism, heterogeneity by using a flexible naming and addressing scheme, application-specified useful life indications on messages, and hop-by-hop authentication and access control. Ongoing development of the DTN architecture and its reference implementation is being undertaken through IRTF's Delay Tolerant Networking Research Group (DTNRG) [1].

In the next sections, we present an overview of the relevant parts of the DTN architecture with an intention of influencing the design of an evolved standard messaging architecture that could form the basis for most of today's popular messaging applications. Some of the DTN mechanisms are clearly not new, such as message-oriented transfer. Techniques such as this have already been employed in most other messaging systems, but these existing sytems were generally not designed for operation on challenged internetworks. While we focus on the application to asynchronous messaging, where latency is not of primary concern, we believe the techniques used to combat intermittent connectivity will not pose too much of a burden in systems that in fact lack such impediments. For a more detailed introduction to the DTN architecture, including classes of service, security and routing, please refer to [8] and [6].

A. Asynchronous Message Delivery

The DTN architecture builds upon the abstraction of reliable asynchronous communication of variable-length, application-specified messages. An asynchronous communication service

encourages applications to not make inappropriate assumptions about the timeliness of responses from their communication peers, allowing the network to queue data for extended periods of time, if necessary, without adversely affecting application operations. In addition, applications may obtain some benefits by choosing their own unit of reliability (so-called application data units) [7].

In a typical implementation, applications register their interest in DTN-level names (see below). Such registrations of interest are held in persistent storage by a local routing agent, so that communications can persist across system restarts or long-term interruptions. When sending, an application may be provided some hint as to how long its sending requests will remain queued locally until they are forwarded to another DTN node, if such information is known ahead of time to the implementation. Such scheduling information can be used, for example, in providing information to the user or to determine the appropriate interval to maintain cache consistency.

B. Routing and Fragmentation

As described above, asynchronous messaging is helpful in tolerating delays and may be helpful to applications in implementing their error handling procedures. It can also be helpful to routing in the network infrastructure: in particular, a message-oriented architecture is able to provide the network routing and scheduling algorithms with *a priori* knowledge of the size and performance requirements of requested data transfers. When there is a significant amount of queuing that can occur prior to transmission over an outbound route (as is the case in the DTN version of store-and-forward), this information can help to optimize scheduling and route selection [9].

The architecture supports *proactive* and *reactive* fragmentation for handling interruptions. In proactive fragmentation, if communication interruptions are known in advance, queued messages can be split into appropriate-sized segments ahead of time and when a communication opportunity becomes available, exactly the correct quantity of data is transferred. When unexpected failures are encountered, DTN employs reactive fragmentation, which essentially amounts to re-packaging data received across a link so that it may be delivered as an independent fragment. Fragments are eventually re-assembled by the final receiver.

Using store-and-forward, DTN routing computations generally take place over a time-evolving graph where a source and sink may never have a contemporaneous end-to-end path available. It utilizes known schedules of link availability (e.g. the tracking patterns of low-earth orbiting satellites) to compute efficient data path selection [9]. DTN routing is also designed to support *opportunistic links* and *probabilistic routing*. Opportunistic links refers to cases in which links become available without any previous knowledge about them (e.g. a PDA comes into range and is willing to route or ferry data). Probabilistic routing refers to cases in which links are not known to deliver messages to their destinations with high probability. Clearly, opportunistic links may require probabilistic routing. In such cases, it may be important to introduce redundancy either by simply replicating message fragments, or using a more sophisticated coding technique such as erasure coding [10] to enhance the probability of eventual delivery of the entire message by employing multiple delivery paths. This area is a current focus of DTNRG.

C. Reliability

The DTN architecture includes several mechanisms for enhancing reliability: an in-network (overlay) hop-by-hop retransmission procedure called *custody transfer*, use of the underlying point-to-point reliable delivery mechanisms offered by the underlying protocols, if available, and also an optional end-to-end acknowledgement mechanism which applications can use to create their own customized error control procedures.

Custody transfer is the acknowledged delivery of a message from one DTN node to another, but need not be utilized across every hop. The node to which custody has been transferred assumes responsibility for eventual reliable delivery, effectively moving the source of the message to the new custodian. The custody transfer mechanism may be used in conjunction with routing to improve reliability by arranging in-transit messages to preferentially be stored at designated custodians (nodes expected to have highly reliable persistant storage, power, etc). To facilitate this behavior, data routing including not only routing to destinations but also routing to custodians is likely to be required, and future work for DTNRG is to investigate the required algorithms to achieve this combination, based upon the initial results in [9].

D. Naming

DTN identifies entities in the network using a *tuple* consisting of a globally unique *region identifier*, which acts as a routing hint, and a region-local *administrative ID*. The region identifier acts like a namespace identifier, and may be drawn from the Internet DNS namespace[1]. Both the region and administrative IDs are variable-length. The administrative ID is only resolved (if required by the underlying protocols) by DTN routing agents sharing the same region identifier as the intended recipient of a message. It is treated as an opaque value by DTN routing otherwise.

The separation of the routing and administrative identifiers allows for the re-use of node administrative identifiers in different regions, and allows for the expression of alternative naming schemes that may not yet be developed. It makes no assumptions about the nature of the name or address used inside the administrative portion (e.g. it is equally able to be an IP, IEEE 802 address, or GSM phone number). At present, DTNRG is focusing on using a URI-compatible structure for encoding such names.

By requiring administrative IDs to only be interpreted by DTN agents sharing the destination region ID, a form of *late binding* is enabled, which can be employed to cause name-to-address mapping to be executed only by nodes that

[1]Using the DNS namespace does not imply the use of DNS mechanisms to resolve such names. For DTN, name-based routing may be used; a DNS-style distributed query does not, in general, need to be performed.

are topologically close to the destination (provided regions are assigned in a topologically-sensitive way). This is to be contrasted with the *early* binding performed by the DNS operation typically executed by today's Internet applications. For Internet DNS, a complete round-trip query must typically be performed in order to resolve a name-to-address mapping *prior* to other communications. For challenged internetworks, this may lead to unacceptable performance.

E. Convergence Layers for Protocol Adaptation

DTN utilizes a protocol known as *bundling* [12] to move messages or message fragments among DTN routing agents. DTN moves *bundles* (the DTN unit of message transfer) through an overlay comprising a set of DTN routing agents located at appropriate points in the underlying network topology, in the same way Internet MTAs move e-mail. DTN routing agents utilize the transport layer protocols of the subnetworks they interconnect[2], and may augment these protocols for delivering bundles, as required. As an example, an Internet transfer using TCP may require the addition of message boundaries, whereas Internet operations using SCTP may not require such augmentation.

The abstraction for adapting lower-layer protocols is known as a *convergence layer*. These (possibly thin) protocol layers adapt the DTN message-oriented routing function for use on the data plane of underlying protocols, and may be required to act somewhat as a session-layer protocol, establishing or clearing connections as required by the DTN routing agent. Convergence layers are specific to the protocols they augment, and affect not only the data plane but the management plane. In particular, events such as "connection established" or "connection dropped" are required by the DTN routing agent when making its path selection and proactive fragmentation decisions.

F. Class of Service

The DTN architecture provides an option for sending applications to supply an abstract class of service (CoS, presently one of four relative levels), and a useful life designation. The CoS designation is used primarily in scheduling decisions, when more than one message is queued awaiting the next available communications opportunity. The CoS designations are advisory in the sense that no end-to-end guarantee is made by the network in delivering messages according to the priority order. However, when a single DTN routing agent is faced with deciding which messages to forward next, it should in general prefer to send those messages of higher priority. An admission control scheme, which may limit the number of highest-priority messages, could be implemented by a network operator, but designing the details of such an approach are currently beyond the scope of DTNRG.

The useful life indicator of a message, supplied by an application and subject to modification by network operator policy, indicates the amount of time beyond the present time the message is still considered to be useful. The overall DTN approach requires time to be synchronized, at least loosely, for this purpose.[3] The bundling protocol presently specifies an originating timestamp that indicates the time a message was initially sent, and a (positive) time delta to the originating timestamp indicating the number of seconds beyond the current time until the message can be safely discarded. The originating timestamp is used both to know when data was sent (by a receiving application), but also as a key to reassemble fragments, in a way similar to the IPv4 fragment ID field.

G. Security

As mentioned in Section I-B, many security proposals involve protocols that require numberous round-trip exchanges between parties and some trusted third party, or require comparatively large authentication credentials to be exchanged prior to initiating communications. When attempting to operate over challenged internetworks, many of these protocols do not work well, as they are unable to contact the server of interest, or they are unable to have access to connectivity for a long enough period of time to transfer the required key material.

As a current area of active work, DTNRG is investigating recently-proposed mechanisms based on Identity Based Cryptography (IBC) [4]. IBC systems, in effect, provide many of the benefits of public key cryptography, but reduce the overhead involved in obtaining and verifying public keys. A public key for an ID string can effectively be formed using only a local funtion of the string and a set of public system parameters. This has potential advantages over conventional public-key cryptography in that public key certificates need not be obtained and transmitted. Although IBC suffers some drawbacks (the conventional IBC scheme typically requires receivers to communicate with a server), it appears that simply pre-keying some nodes with their own private keys may offer reasonably efficient operation over disruption-prone networks at an acceptable security risk.

III. Comparison with Existing Systems

A. Internet E-Mail

As an existing asynchronous communication mechanism, electronic mail possesses many properties in common with DTN (indeed, it would be fair to say DTN possesses many of the properties of e-mail). However, there are some differences worth noting. First, e-mail (at least present Internet e-mail) employs only a very primitive form of routing (MX records). It depends primarily on the underlying IP routing to achieve its delivery. In addition, it delivers mail from one MTA to another and contains no provision for handling *opportunistic* communication opportunities (e.g. those made available by a nearby PDA that is willing to 'ferry' your e-mail for you to its home base). It also makes no use of replication or modern coding techniques (cf. erasure codes) in dealing with MTAs that may not be able to eventually deliver mail to its destination

[2]Actually, a transport layer is not strictly required; the DTN routing agent is layer-agnostic.

[3]Synchronized time is also used by the routing algorithms that operate on known schedules of time-evolving topology graphs.

with high probability. It generally does not employ hop-by-hop authentication, although some approaches resembling such functionality are being undertaken presently [11], [2]

With respect to its use of underlying transport protocols, typical e-mail today runs atop the TCP transport protocol, which works admirably well in wired networks or even wireless networks that are not subject to significant disruption. When TCP is required to run over intermittent links, it performs poorly, and a number of mitigating devices (PEPs–Performance Enhancing Proxies) have been devised to help the situation. These devices are not without problems; in particular they do not work effectively when mechanisms such as IPSec are employed or when IP level routing is asymmetric [5].

The DTN architecture is well-equipped to support the natural asynchronous mode communication in e-mail and may improve its performance by providing the proactive and reactive fragmentation capabilities, as well as the capability to incorporate multi-path routing. Such benefits could be felt most strongly in wireless and other disruption-prone networks.

B. Network News/NNTP

As another popular asynchronous messaging service, network news has been used for many years, including situations in which "always-on" connectivity was not available. News is a "pull" model whereas electronic mail and DTN are fairly characterized as "push" models. More specifically, sending news agents publish only to multicast groups (newsgroup names), and not to individuals. It provides only limited feedback with respect to acknowledgments that news has been delivered. However, it shares many of the same issues with respect to store-and-forward with e-mail and DTN.

The NNTP routing graph is in general not dynamic, and instead relies on some form of underlying routing files, set up by hand, to determine nodes to which it should supply news traffic. As DTN could be used for e-mail it could also be used for carrying news (or vice-versa). If used as the underlying transport, it would offer disruption tolerance and security without significant extensions.

C. SMS/MMS

SMS and MMS messaging are the services available on cellular telephone systems providing the capability to deliver asynchronous text or rich content to other cellular telephone users, respectively. These networks operate separately from the Internet, and are sometimes considerd to be more secure, accessible, and less open to misuse (such as spam). Most cellular systems deliver these types of messages with small delay (on the order of a second or less), so they can be used in a quasi-synchronous way. However, when a recipient is not available to receive an incoming message, it is queued until read. Pricing has had a profound effect on the methods of usage. In particular, for users with effectively flat-rate calling plans it is often much more convenient to simply place a cellular phone call because of the inconvenience imposed by the user interface provided on most cellular phones. Text messaging is therefore used most often either when one of the two communicating parties is not present and will not likely be reachable by e-mail, or when text messages are cheap in comparison with cellular phone calls.

These sytems frequently contain one or more Internet gateways, allowing Internet e-mail and a provider's messaging system to be interconnected in a controlled way. The provider may elect to filter certain messages, and may employ differential pricing depending on the types or lengths of messages delivered. The addressing format is somewhat different from Internet mail, so gateways used to interconnect the systems must perform some form of address translation (frequently, just the phone number encoded as the user name in an Internet-style e-mail address).

DTN provides similar functions to these, including store-and-forward as well as the translating/gateway function required in SMS/e-mail gateways. It provides a similar naming capability: the `<phone>@service-provider` syntax effectively equates to the {`region, admin-ID`} syntax of DTN names.

D. Instant Messaging

Instant messaging refers to messages delivered over Internet clients in relatively short times, typically using TCP/IP with a centralized server. IMs are generally not stored, so an intended recipient that is not present will not receive the message. IM fills the void between an e-mail and a telephone call by providing some indication of presence, and some way to interrupt a user for a quick question/response.

To be effective for this purpose, IM systems require limited delay, and thus a connected or mostly-connected network infrastructure. DTN is focused on tolerating networks which are often not connected, so the match may not seem appropriate. However, even for networks that are operating properly most of the time, the effects of short-term disruption can be severe for protocols such as TCP, so some mechanism for re-attachment is generally required. While this certainly can be implemented at the application layer, basing such a system on DTN (which might be present for supporting the other message-oriented applications) could save some modest complexity inside applications.

IV. Conclusion

In this paper, we consider the features, services, and usage models of a number of popular messaging systems with the purpose of creating a new common messaging architecture. While most of these existing systems operate well on their own, they require specialized proxies to interoperate, and frequently fail when faced with operating over challenged internetworks. With the increase in popularity of wireless networks, the importance of handling network disruptions appears to be increasing, so it becomes important to explore the robustness of messaging systems with respect to poor network performance.

In reviewing the usage models of existing messaging systems in conjunction with underlying network performance, we find that some messaging paradigms, in particular those that are asynchronous in nature, are more easily mapped to networks that suffer from disruption. Thus, it would appear

the most appropriate basic abstraction for a messaging service is an asynchronous message, delivered *eventually* to its destination.

Based on our previous work on asynchronous messaging for challenged internetworks, we suggest the Delay Tolerant Networking (DTN) architecture as a useful basis for the design of a common future messaging system that is expected to operate over networks prone to disruption. The architecture supports conventional routing as well as data mules, pre-scheduled and opportunistic network links, extreme network heterogeneity, and reasonable performance over links suject to unexpected disruption. It has an active and growing community of researchers in the IRTF Delay Tolerant Networking Research Group, a has also become a driving influence for the recently announced DARPA program in Disruption Tolerant Networks.

REFERENCES

[1] Delay tolerant networking research group. See http://www.dtnrg.org.

[2] Sender policy framework. See http://spf.pobox.com.

[3] The wizzy digital courier. Available at http://wizzy.org.za/article/articlestatic/19/1/2/.

[4] D. Boneh and M. Franklin. Identity based encryption from the weil pairing. In *SIAM J. of Computing 2003*, 2001.

[5] J. Border, M. Kojo, J. Griner, G. Montenegro, and Z. Shelby. Performance enhancing proxies intended to mitigate link-related degradations. 2001.

[6] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss. Delay tolerant network architecture. *Internet Draft draft-irtf-dtnrg-arch-02.txt (work in progress)*, July 2004.

[7] D. Clark and D. Tennenhouse. Architectural considerations for a new generation of protocols. In *Proc. SIGCOMM*, Aug. 1990.

[8] K. Fall. A Delay-Tolerant Network Architecture for Challenged Internets. In *Proc. SIGCOMM 2003*, Aug. 2003.

[9] S. Jain, K. Fall, and R. Patra. Routing in a Delay Tolerant Network. In *Proc. SIGCOMM 2004*, Aug. 2004.

[10] M. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. Spielman. Efficient erasure correcting codes. In *Proc. IEEE Trans. on Information Theory*, Feb. 2001.

[11] J. Lyon and M. Wong. Sender id: Authentication e-mail. *Internet Draft draft-ietf-marid-core-03.txt (work in progress)*, Aug. 2004.

[12] K. Scott and S. Burleigh. Bundle protocol specification. *Internet Draft draft-irtf-bundle-spec-02.txt (work in progress)*, Sept. 2004.

[13] R. C. Shah, S. Roy, S. Jain, and W. Brunette. Data MULEs: Modeling a Three-tier Architecture for Sparse Sensor Networks. In *Proceedings of the IEEE Workshop on Sensor Network Protocols and Applications*, May 2003.